R E P U B L I Q U E F R A N Ç A PCT/EP2004/051411

INPI
INSTITUT NATIONAL DE LA PROPRIETE INDUSTRIELLE

REC'D 06 SEP 2004
WIPO PCT

05 AUG 2004

BREVET D'INVENTION

CERTIFICAT D'UTILITÉ - CERTIFICAT D'ADDITION

COPIE OFFICIELLE

Le Directeur général de l'Institut national de la propriété industrielle certifie que le document ci-annexé est la copie certifiée conforme d'une demande de titre de propriété industrielle déposée à l'Institut.

Fait à Paris, le 29 JUIL. 2004

Pour le Directeur général de l'Institut
national de la propriété industrielle
Le Chef du Département des brevets

Martine PLANCHE

PRIORITY DOCUMENT
SUBMITTED OR TRANSMITTED IN COMPLIANCE WITH RULE 17.1(a) OR (b)

INSTITUT NATIONAL DE LA PROPRIETE INDUSTRIELLE
SIEGE
26 bis, rue de Saint-Petersbourg
75800 PARIS cedex 08
Téléphone : 33 (0)1 53 04 53 04
Télécopie : 33 (0)1 53 04 45 23
www.inpi.fr

INPI
INSTITUT NATIONAL DE LA PROPRIÉTÉ INDUSTRIELLE
26 bis, rue de Saint Pétersbourg
75800 Paris Cedex 08
Téléphone : 33 (1) 53 04 53 04 Télécopie : 33 (1) 42 94 86 54

BREVET D'INVENTION
CERTIFICAT D'UTILITÉ

Code de la propriété intellectuelle - Livre VI

cerfa N° 11354*02

REQUÊTE EN DÉLIVRANCE
page 1/2

BR1

Cet imprimé est à remplir lisiblement à l'encre noire GB 540 @ W / 010801

Réservé à l'INPI

REMISE DES PIÈCES
DATE 31 JUIL 2003
LIEU 75 INPI PARIS
N° D'ENREGISTREMENT NATIONAL ATTRIBUÉ PAR L'INPI 0309457
DATE DE DÉPÔT ATTRIBUÉE PAR L'INPI 31 JUIL. 2003

Vos références pour ce dossier *(facultatif)* 016870 GEM1382

1 NOM ET ADRESSE DU DEMANDEUR OU DU MANDATAIRE À QUI LA CORRESPONDANCE DOIT ÊTRE ADRESSÉE

Cabinet BALLOT

122, rue Edouard Vaillant
92593 LEVALLOIS-PERRET Cedex

Confirmation d'un dépôt par télécopie ☐ N° attribué par l'INPI à la télécopie

2 NATURE DE LA DEMANDE	Cochez l'une des 4 cases suivantes
Demande de brevet	☒
Demande de certificat d'utilité	☐
Demande divisionnaire	☐
Demande de brevet initiale	N° Date
ou demande de certificat d'utilité initiale	N° Date
Transformation d'une demande de brevet européen *Demande de brevet initiale*	☐ N° Date

3 TITRE DE L'INVENTION (200 caractères ou espaces maximum)

PROCEDE POUR LA MISE EN ŒUVRE SECURISEE D'UN ALGORITHME DE CRYPTOGRAPHIE DE TYPE RSA ET COMPOSANT CORRESPONDANT

4 DÉCLARATION DE PRIORITÉ OU REQUÊTE DU BÉNÉFICE DE LA DATE DE DÉPÔT D'UNE DEMANDE ANTÉRIEURE FRANÇAISE	
	Pays ou organisation Date N°
	Pays ou organisation Date N°
	Pays ou organisation Date N°
	☐ **S'il y a d'autres priorités, cochez la case et utilisez l'imprimé «Suite»**

5 DEMANDEUR (Cochez l'une des 2 cases)		☒ Personne morale ☐ Personne physique
Nom ou dénomination sociale		GEMPLUS
Prénoms		
Forme juridique		S.A. (Société Anonyme)
N° SIREN		3 4 9 7 1 1 2 0 0
Code APE-NAF		3 2 1 B
Domicile ou siège	Rue	Avenue du Pic de Bertagne Parc d'activités de Gémenos
	Code postal et ville	1 3 4 2 0 GEMENOS
	Pays	France
Nationalité		Française
N° de téléphone *(facultatif)*		N° de télécopie *(facultatif)*
Adresse électronique *(facultatif)*		
		☐ **S'il y a plus d'un demandeur, cochez la case et utilisez l'imprimé «Suite»**

Remplir impérativement la 2ème page

INPI
INSTITUT NATIONAL DE LA PROPRIETE INDUSTRIELLE

BREVET D'INVENTION
CERTIFICAT D'UTILITÉ

REQUÊTE EN DÉLIVRANCE
page 2/2

BR2

Réservé à l'INPI

REMISE DES PIÈCES
DATE 31 JUIL 2003
LIEU 75 INPI PARIS

N° D'ENREGISTREMENT NATIONAL ATTRIBUÉ PAR L'INPI 0309457

DB 540 W / 210502

6 MANDATAIRE *(s'il y a lieu)*		016870 GEM1382
Nom		BENTZ
Prénom		Jean-Paul
Cabinet ou Société		CABINET BALLOT
N °de pouvoir permanent et/ou de lien contractuel		
Adresse	Rue	122, rue Edouard Vaillant
	Code postal et ville	92593 Levallois-Perret Cedex
	Pays	FRANCE
N° de téléphone *(facultatif)*		01 49 64 61 00
N° de télécopie *(facultatif)*		01 49 64 61 20
Adresse électronique *(facultatif)*		
7 INVENTEUR (S)		**Les inventeurs sont nécessairement des personnes physiques**
Les demandeurs et les inventeurs sont les mêmes personnes		☐ Oui ☒ Non : **Dans ce cas remplir le formulaire de Désignation d'inventeur(s)**
8 RAPPORT DE RECHERCHE		**Uniquement pour une demande de brevet (y compris division et transformation)**
Établissement immédiat ou établissement différé		☒ ☐
Paiement échelonné de la redevance *(en deux versements)*		**Uniquement pour les personnes physiques effectuant elles-mêmes leur propre dépôt** ☐ Oui ☐ Non
9 RÉDUCTION DU TAUX DES REDEVANCES		**Uniquement pour les personnes physiques** ☐ Requise pour la première fois pour cette invention *(joindre un avis de non-imposition)* ☐ Obtenue antérieurement à ce dépôt pour cette invention *(joindre une copie de la décision d'admission à l'assistance gratuite ou indiquer sa référence)* : **AG** ⌊⌊⌊⌊⌊
10 SÉQUENCES DE NUCLEOTIDES ET/OU D'ACIDES AMINÉS		☐ Cochez la case si la description contient une liste de séquences
Le support électronique de données est joint		☐
La déclaration de conformité de la liste de séquences sur support papier avec le support électronique de données est jointe		☐
Si vous avez utilisé l'imprimé «Suite», indiquez le nombre de pages jointes		

11 SIGNATURE DU DEMANDEUR OU DU MANDATAIRE
(Nom et qualité du signataire)

Jean-Paul BENTZ
99-0308

VISA DE LA PRÉFECTURE OU DE L'INPI
L. GUICHET

La loi n°78-17 du 6 janvier 1978 relative à l'informatique, aux fichiers et aux libertés s'applique aux réponses faites à ce formulaire. Elle garantit un droit d'accès et de rectification pour les données vous concernant auprès de l'INPI.

PROCEDE POUR LA MISE EN ŒUVRE SECURISEE D'UN ALGORITHME DE CRYPTOGRAPHIE DE TYPE RSA ET COMPOSANT CORRESPONDANT

La présente invention se rapporte à un procédé pour la mise en œuvre sécurisée d'un algorithme de cryptographie dans un composant électronique et, plus particulièrement, pour la mise en œuvre sécurisée d'un algorithme de cryptographie de type RSA.

L'invention concerne également le composant électronique correspondant.

De tels composants sont notamment utilisés dans des applications où l'accès à des services ou à des données est sévèrement contrôlé.

Ils ont une architecture dite logicielle, c'est-à-dire programmable formée autour d'un microprocesseur et de mémoires, dont une mémoire programme non volatile de type *EEPROM* qui contient un ou plusieurs nombres secrets. Il s'agit d'une architecture généraliste apte à exécuter n'importe quel algorithme.

Ces composants sont utilisés dans des systèmes informatiques, embarqués ou non. Ils sont notamment utilisés dans les cartes à puce, pour certaines applications de celles-ci. Ce sont par exemple des applications d'accès à certaines banques de données, des applications bancaires, des applications de télépéage, par exemple pour la télévision, la distribution d'essence ou encore le passage de péages d'autoroutes.

Ces composants ou ces cartes mettent donc en œuvre un algorithme de cryptographie pour assurer le chiffrement de données émises et/ou le déchiffrement de données reçues, l'authentification ou la signature numérique d'un message.

A partir de ce message appliqué en entrée à la carte par un système hôte (serveur, distributeur bancaire...) et de nombres secrets contenus dans la carte, la carte fournit en retour au système hôte ce message chiffré, authentifié ou signé, ce qui permet par exemple au système hôte d'authentifier le composant ou la carte, d'échanger des données...

Les caractéristiques des algorithmes de cryptographie peuvent être connues : calculs effectués, paramètres utilisés. La seule inconnue est le ou les nombres secrets. Toute la sécurité de ces algorithmes de cryptographie tient dans ce(s) nombre(s) secret(s) contenu(s) dans la carte et inconnu(s) du monde extérieur à la carte. Ce nombre secret ne peut être déduit de la seule connaissance du message appliqué en entrée et du message chiffré fourni en retour.

Or, il est apparu que des attaques externes basées sur des grandeurs physiques mesurables à l'extérieur du composant lorsque celui-ci est en train de dérouler l'algorithme de cryptographie, permettent à des tiers mal intentionnés de trouver le(s) nombre(s) secret(s) contenu(s) dans cette carte. Ces attaques sont appelées attaques à canaux cachés (« Side channel attacks » en anglais) ; on distingue parmi ces attaques à canaux cachés, les attaques SPA, acronyme anglo-saxon pour *Single Power Analysis* basées sur une voire quelques

mesures et les attaques DPA, acronyme anglo-saxon pour *Differential Power Analysis* basées sur des analyses statistiques issues de nombreuses mesures. Le principe de ces attaques à canaux cachés repose par exemple sur le fait que la consommation en courant du microprocesseur exécutant des instructions varie selon l'instruction ou la donnée manipulée.

Il existe également un type d'attaque, dite « attaque par faute ». Dans ce type d'attaque, l'attaquant injecte une faute quelconque pendant le calcul d'un algorithme cryptographique, dans le but d'exploiter la présence de cette faute pour extraire une information secrète.

La faute peut aussi provenir d'une erreur de calcul due au matériel mettant en œuvre l'algorithme cryptographique. On considère néanmoins, dans un cas comme dans l'autre, qu'il s'agit d'une attaque par faute.

Ces différents types d'attaque sont notamment envisageables avec les algorithmes de cryptographie à clé publique comme par exemple l'algorithme RSA (du nom de ses auteurs Rivest, Shamir, Adleman), qui est celui le plus utilisé en cryptographie dans ce domaine d'application, et auquel la présente invention s'applique plus particulièrement.

On rappelle ci-après brièvement les principales caractéristiques du *système cryptographique* à clé publique RSA.

La première réalisation de schéma de chiffrement *et de signature* à clé publique fut mise au point en 1977 par Rivest, Shamir et Adleman, qui ont inventé le

système *cryptographique* RSA. La sécurité de RSA repose sur la difficulté de factoriser un grand nombre qui est le produit de deux nombres premiers. Ce système est le système *cryptographique* à clé publique le plus utilisé. Il peut être utilisé comme procédé de chiffrement ou comme procédé de signature.

Le principe du système *cryptographique* RSA est le suivant. Il consiste d'abord à générer la paire de clés RSA.

Ainsi, chaque utilisateur crée une clé publique RSA et une clé privée correspondante, suivant le procédé suivant en 5 étapes :

1) Générer deux nombres premiers distincts p et q;

2) Calculer $n=pq$ et $\Phi(n)=(p-1)(q-1)$, Φ étant appelée la fonction indicatrice d'Euler;

3) *Sélectionner un entier e, $1<e<\Phi(n)$, tel que $pgcd(e,\Phi(n))=1$, aléatoirement ou au choix de l'utilisateur qui pourrait donc choisir e petit tel que $e = 2^{16}+1$ ou $e = 3$ ou $e = 17$;*

4) Calculer l'unique entier d, $1<d<\Phi(n)$, tel que :

$$e.d=1 \text{ modulo } \Phi(n) \; ; \qquad (1)$$

5) La clé publique est (n,e); la clé privée est d ou (d,p,q).

Les entiers e et d sont appelés respectivement *exposant public et exposant privé*. L'entier n est appelé le module RSA.

Une fois les paramètres publics et privés définis, étant donné x, avec $0<x<n$, l'opération publique sur x qui peut être par exemple le chiffrement du message x consiste à calculer : $y = x^e \text{ modulo } n$ (2)

Dans ce cas, l'opération privée correspondante est l'opération de déchiffrement du message chiffré y, et consiste à calculer :

$$y^d \text{ modulo } n \qquad (3)$$

L'opération publique sur x peut encore être la vérification de la signature x, et consiste à calculer : $y = x^e$ modulo n (2)

L'opération privée correspondante est alors la génération d'une signature x à partir du message préalablement encodé y par application d'une fonction de hachage μ ("padding" selon la terminologie anglo-saxonne), et consiste à calculer :

$$y^d \text{ modulo } n \qquad (3)$$

Avec $x = y^d$ modulo n puisque $e.d = 1$ modulo $\Phi(n)$

On va présenter un autre mode de fonctionnement dit mode CRT car basé sur le théorème des restes chinois (« Chinese Remainder Therorem » ou CRT en anglais) et quatre fois plus rapide que celui de l'algorithme RSA standard. Selon ce mode CRT, on n'effectue pas directement les calculs modulo n mais on effectue d'abord les calculs modulo p et modulo q.

Les paramètres publics sont (n, e) mais les paramètres privés sont dans ce mode (p, q, d) ou (p, q, d_p, d_q, i_q) avec

$d_p = d$ modulo$(p-1)$, $d_q = d$ modulo$(q-1)$

et $i_q = q^{-1}$ modulo p

Par la relation (1), on obtient, :

$$ed_p = 1 \text{ modulo}(p-1) \text{ et } ed_q = 1 \text{ modulo}(q-1) \qquad (4)$$

L'opération publique s'effectue de la même façon que pour le mode de fonctionnement standard. Par contre, pour l'opération privée, on calcule d'abord :

$x_p = y^{dp}$ modulo p et $x_q = y^{dq}$ modulo q

Ensuite, par application du théorème des restes chinois, on obtient $x = y^d$ modulo n par :

$$x = CRT(x_p, x_q) = x_q + q[i_q(x_p - x_q) \text{ modulo } p] \quad (5)$$

Une orientation importante dans le domaine de la cryptographie à clé publique utilisant le schéma de chiffrement RSA consiste donc à sécuriser la mise en œuvre des algorithmes RSA contre les différents types d'attaques possibles évoqués plus haut, notamment les attaques à canaux cachés telles que les attaques DPA et SPA, ainsi que les attaques dites par faute où l'attaquant, par une méthode quelconque, injecte une faute pendant le calcul d'une opération privée de l'algorithme RSA, dans le but d'obtenir une valeur corrompue à partir de laquelle il est possible, dans certains cas, de déduire certaines données secrètes.

Dans l'état de la technique, certains procédés de contre-mesure ont été envisagés pour parer ces différents types d'attaques.

Notamment, une contre-mesure possible pour parer les attaques de type DPA *(et SPA)* contre le RSA en mode standard consiste à *rendre aléatoire* le calcul de l'opération privée du RSA (signature ou déchiffrement) en introduisant dans le calcul une valeur aléatoire.

Ainsi, une méthode de contre-mesure de ce type consiste à calculer l'opération privée en mode standard (3) $x = y^d$ modulo n de la façon suivante :

$x = y^{d-r} . y^r$ modulo n, avec r étant un nombre entier aléatoire. Toutefois, l'inconvénient de cette

méthode de contre-mesure est que le temps de calcul est doublé.

Une autre méthode de contre-mesure de ce type pour parer les attaques DPA *(et SPA)* contre le RSA en mode standard consiste à calculer l'opération privée (3) $x = y^d$ modulo n de la façon suivante :

$x = y^{(d+r.\Phi(n))}$ modulo n, avec r un entier aléatoire. Cependant, un inconvénient de cette méthode est qu'elle requiert la connaissance de la valeur de $\Phi(n)$, qui est généralement inconnue par l'algorithme de cryptographie qui met en œuvre l'opération privée (signature ou déchiffrement).

Aussi, une variante de cette méthode est proposée, basée non plus sur la connaissance de la valeur de $\Phi(n)$ mais sur celle de la valeur de *l'exposant public* e. En effet, on a d'après (1) : $e.d = 1$ modulo $\Phi(n)$, aussi, il existe un entier k tel que : $e.d-1 = k.\ \Phi(n)$;

En conséquence, l'expression $x = y^{(d+r.\Phi(n))}$ modulo n peut se calculer sous la forme :

$x = y^{(d+r.(ed-1))}$ modulo n, avec r un entier aléatoire.

Cette méthode de contre-mesure est donc calculatoirement équivalente à celle dont elle découle, avec l'avantage cependant de ne pas nécessiter la connaissance de la valeur de $\Phi(n)$. Elle requiert moins de mémoire en ce sens qu'elle ne nécessite pas de garder $\Phi(n)$.

Toutefois, cette variante de contre-mesure, pour pouvoir être mise en œuvre, nécessite d'avoir la connaissance de la valeur de l'exposant public e. Or, dans de nombreuses applications de cryptographie, le

composant ou le dispositif mettant en œuvre l'opération privée de l'algorithme RSA ne dispose pas toujours de *l'exposant public* e, notamment lorsqu'il n'exécute que l'opération privée. *L'exposant public* e est donc dans ce contexte généralement inconnu ou indisponible.

Les contre-mesures décrites précédemment sont principalement destinées à parer les attaques de type DPA. Cependant, elles rendent également plus difficiles les attaques de type SPA dans la mesure où l'exécution de l'algorithme est non-déterministe.

Pour ce qui est de l'autre type d'attaque qui a été évoqué, dite attaque par faute, la meilleure protection possible pour la parer consiste à tester, en mode standard, que la valeur x obtenue par application de l'opération privée vérifie effectivement la relation $x^e = y$ modulo n de l'opération publique. Si ce n'est pas le cas, on ne retournera pas la valeur y pour éviter son utilisation à des fins de cryptanalyse.

En mode CRT, la protection consiste à vérifier d'une part, si effectivement les relations $x^e = y$ modulo p et, d'autre part, $x^e = y$ modulo q sont vérifiées.

En effet, lorsque ces relations sont vérifiées, on est assuré qu'il n'y a pas eu d'erreurs pendant le déroulement de l'opération privée de l'algorithme RSA.

Toutefois, un inconvénient empêchant la mise en œuvre de telles vérifications contre les attaques par faute, en mode standard ou en mode CRT, est que ces opérations de vérification nécessitent également la connaissance préalable de *l'exposant public* e. Or, comme déjà vu, le composant ou le dispositif mettant en

œuvre l'opération privée de l'algorithme RSA, en mode standard ou CRT, ne dispose pas toujours de *l'exposant public* e, notamment lorsqu'il n'exécute que l'opération privée. *L'exposant public* e est donc dans ce contexte généralement inconnu ou indisponible.

Le document de brevet FR 2 830 146 (D1) propose à cet effet un procédé permettant de réaliser certaines étapes d'un algorithme de cryptographie, et notamment de type RSA en mode standard ou CRT, utilisant un *exposant public* e que l'on ne connaît pas a priori.

Le procédé objet de D1 permet en particulier de réaliser une contre-mesure, notamment aux attaques par faute, qui offre la meilleure protection possible telle qu'évoquée ci-dessus, même lorsqu'on ne connaît pas *l'exposant public* e.

Pour ce faire, soit (e, d) une paire correspondante d'exposants RSA respectivement *public et privé* et soit n le module RSA. D1 part de la constatation suivante selon laquelle dans 95% des cas, la valeur de *l'exposant public* e est choisie parmi les valeurs $2^{16}+1$, 3, 17. La méthode de D1, exposée brièvement ici en référence au mode standard mais qui peut tout autant s'appliquer au mode CRT, consiste alors à vérifier que e est bien égal à une de ces valeurs en testant successivement si $e_i.d = 1$ modulo $\Phi(n)$, avec $e_i \in E = \{2^{16}+1, 3, 17\}$, jusqu'à ce que la relation soit vérifiée.

Lorsque la relation est vérifiée pour un e_i, alors on sait que $e=e_i$. Une fois la valeur de *l'exposant public* e déterminée de cette façon, e est mémorisée en vue de son utilisation dans des calculs de l'algorithme

RSA visant à vérifier qu'il n'y a pas eu d'erreurs, dues à une attaque par faute, pendant le déroulement d'une opération privée correspondante de l'algorithme RSA. Ainsi, connaissant e, il est possible d'affirmer avec une probabilité égale à 1 que l'opération privée se rapportant par exemple à la génération d'une signature s, avec $s = \mu(m)^d$ modulo n, $\mu(m)$ étant la valeur obtenue par l'application d'une fonction μ de padding au message m à signer, a été effectuée sans erreur en vérifiant simplement que la valeur s obtenue vérifie la relation $s^e = \mu(m)$ modulo n de l'opération publique correspondante.

Si aucune valeur de e_i n'a pu être attribuée à e, il convient alors de constater selon D1 que les calculs de l'algorithme RSA utilisant la valeur e pour la sécurisation contre les attaques par faute ne peuvent être effectués.

Cependant, un inconvénient de la méthode proposée par D1 est qu'elle implique d'exécuter une pluralité de calculs modulaires lorsqu'on teste successivement si la relation $e_i d = 1$ modulo $\Phi(n)$ est vérifiée, pour une valeur de e_i parmi les e_i envisagés. Or les calculs modulaires sont des calculs complexes. Cette méthode se révèle donc pénalisante en terme de temps de calcul et de ressources de calcul.

Aussi, le problème qui se pose est de pallier les inconvénients précités.

Plus particulièrement, un but de la présente invention consiste à déterminer d'une façon qui ne soit pas pénalisante en terme de rapidité et de complexité de calcul, la valeur d'un *exposant public* e parmi un

ensemble de valeurs probables prédéterminées, lorsque l'on ne connaît pas cette valeur de e a priori, l'exposant e étant mis en œuvre dans certaines étapes d'un algorithme de cryptographie de type RSA en mode standard ou CRT.

Un autre but consiste donc à pouvoir mettre en œuvre, une fois la valeur *de l'exposant public* e déterminée, des opérations de contre-mesure utilisant la valeur de *l'exposant public* e, visant à parer d'une part, les attaques dites attaques par faute et, d'autre part, les attaques dites à canaux cachés, notamment de type DPA et SPA, susceptibles d'être conduites lors de la mise en œuvre d'une opération privée d'un algorithme de cryptographie, notamment de type RSA.

Avec ces objectifs en vue, l'invention concerne un procédé pour la mise en œuvre sécurisée d'un algorithme de cryptographie à clé publique, *ladite clé publique étant composée d'un nombre entier n, produit de deux grands nombres premiers p et q*, et d'un exposant *public* e, ledit procédé consistant à déterminer un ensemble E comprenant un nombre prédéterminé de valeurs e_i susceptibles de correspondre à la valeur de l'exposant *public* e, les e_i étant des nombres *premiers*, caractérisé en ce qu'il comprend les étapes suivantes consistant à:

a) définir une valeur $\epsilon = \prod_{ei \in E} e_i$

telle que ϵ/e_i soit inférieur à $\Phi(n)$ pour tout e_i appartenant à E, Φ étant la fonction indicatrice d'Euler;

b) appliquer la valeur ϵ dans un calcul prédéterminé;

c) pour chacun des e_i de E, tester si le résultat dudit calcul prédéterminé est égal à une valeur ϵ/e_i:

- si c'est le cas, alors attribuer la valeur e_i à e et mémoriser e en vue de son utilisation dans des calculs dudit algorithme de cryptographie;
- sinon, constater que les calculs dudit algorithme de cryptographie utilisant la valeur e ne peuvent être effectués.

L'avantage est donc clairement que l'on n'ait plus qu'une seule multiplication modulaire.

Dans une première variante, l'algorithme de cryptographie est basé sur un algorithme de type RSA en mode standard.

En rapport avec cette première variante, le calcul prédéterminé de l'étape b) consiste à calculer une valeur C :

C = ϵ.d modulo $\Phi(n)$, d étant la clé privée correspondante de l'algorithme RSA telle que e.d = 1 modulo $\Phi(n)$ et Φ étant la fonction indicatrice d'Euler.

Selon une alternative, le calcul prédéterminé de l'étape b) consiste à calculer une valeur C :

C = ϵ.d modulo $\lambda(n)$, d étant la clé privée correspondante de l'algorithme RSA telle que e.d = 1 modulo $\lambda(n)$ et λ étant la fonction de Carmichael.

Dans une seconde variante, l'algorithme de cryptographie est basé sur un algorithme de type RSA en mode CRT.

En liaison avec cette seconde variante, le calcul prédéterminé de l'étape b) consiste à calculer une valeur C :

$C = \epsilon.d_p$ modulo $(p-1)$, d_p étant la clé privée correspondante de l'algorithme RSA telle que $e.d_p = 1$ modulo $(p-1)$.

Selon une alternative, le calcul prédéterminé de l'étape b) consiste à calculer une valeur C :

$C = \epsilon.d_q$ modulo $(q-1)$, d_q étant la clé privée correspondante de l'algorithme RSA telle que $e.d_q = 1$ modulo $(q-1)$.

Selon une autre alternative, le calcul prédéterminé de l'étape b) consiste à calculer deux valeurs C_1 et C_2 telles que :

$C_1 = \epsilon.d_p$ modulo $(p-1)$, d_p étant la clé privée correspondante de l'algorithme RSA telle que $e.d_p = 1$ modulo $(p-1)$,

$C_2 = \epsilon.d_q$ modulo $(q-1)$, d_q étant la clé privée correspondante de l'algorithme RSA telle que $e.d_q = 1$ modulo $(q-1)$,

et en ce que l'étape de test c) consiste pour chaque e_i, à tester si C_1 *et/ou* C_2 est *égal* à la valeur ϵ/e_i:

- si c'est le cas, alors attribuer la valeur e_i à e et mémoriser e en vue de son utilisation dans des calculs dudit algorithme de cryptographie;

- sinon, constater que les calculs dudit algorithme de cryptographie utilisant la valeur e ne peuvent être effectués.

Selon la première variante et dans le cas où une valeur e_i a été attribuée à e, les calculs utilisant la valeur e consistent à :

-choisir un entier aléatoire r;

-calculer une valeur d^* telle que $d^* = d+r.(e.d-1)$;

-mettre en œuvre une opération privée de l'algorithme dans laquelle une valeur x est obtenue à partir d'une valeur y en appliquant la relation $x = y^{d^*}$ modulo n.

Selon la première variante et dans le cas où une valeur e_i a été attribuée à e, les calculs utilisant la valeur e consistent à obtenir, à l'issue d'une opération privée de l'algorithme, une valeur x à partir d'une valeur y et à vérifier si $x^e = y$ modulo n.

Selon la deuxième variante et dans le cas où une valeur e_i a été attribuée à e, les calculs utilisant la valeur e consistent à obtenir, à l'issue d'une opération privée de l'algorithme, une valeur x à partir d'une valeur y et à vérifier d'une part, si $x^e = y$ modulo p et, d'autre part, si $x^e = y$ modulo q.

De préférence, l'ensemble E comprend au moins les valeurs e_i suivantes 3, 17, $2^{16}+1$.

L'invention concerne également un composant électronique caractérisé en ce qu'il comprend des moyens pour la mise en œuvre du procédé tel que défini précédemment.

L'invention concerne encore une carte à puce comprenant un composant électronique tel que défini.

L'objet de l'invention concerne également un procédé pour la mise en œuvre sécurisée d'un algorithme de cryptographie à clé publique, ladite clé publique *étant composée d'un nombre entier n, produit de deux grands nombres premiers p et q,* et d'un exposant *public* e, ledit procédé consistant à déterminer un ensemble E comprenant un nombre prédéterminé de valeurs e_i susceptibles de correspondre à la valeur de l'exposant *public* e, les e_i étant des nombres *premiers,* caractérisé en ce qu'il consiste à réaliser les étapes suivantes consistant à:

a) choisir une valeur e_i parmi les valeurs de l'ensemble E;

b) *si* $\delta(p)=\delta(q)$, tester si la valeur e_i choisie vérifie la relation : $(1-e_i.d)$ modulo $n < e_i.2^{(\delta(n)/2)+1}$

ou ladite relation simplifiée :

$(-e_i.d)$ modulo $n < e_i.2^{(\delta(n)/2)+1}$

avec $\delta(p)$, $\delta(q)$ et $\delta(n)$ les fonctions donnant le nombre de bits *codant respectivement le nombre p, le nombre q* et le nombre n;

sinon, dans le cas où p et q sont déséquilibrés, tester si la valeur e_i choisie vérifie la relation :

$(1-e_i.d)$ modulo $n < e_i.2^{g+1}$

ou ladite relation simplifiée :

$(-e_i.d)$ modulo $n < e_i.2^{g+1}$

avec $g=max(\delta(p), \delta(q))$, *si* $\delta(p)$ *et* $\delta(q)$ *sont connus ou, dans le cas contraire, avec* $g= \delta(n)/2+t$, *où t désigne le facteur de déséquilibre ou une borne sur ce facteur;*

c) si la relation de test appliquée à l'étape précédente est vérifiée, alors e = e_i, et mémoriser e en vue de son utilisation dans des calculs dudit algorithme de cryptographie,

- si ce n'est pas le cas, réitérer les étapes précédentes en choisissant une autre valeur de e_i dans l'ensemble E jusqu'à ce qu'une valeur de e_i puisse être attribuée à e et si aucune valeur de e_i ne peut être attribuée à e alors constater que les calculs dudit algorithme de cryptographie utilisant la valeur de e ne peuvent pas être effectués.

Le fait de choisir l'ordre des e_i comme celui des probabilités d'apparitions des exposants publics permet de gagner du temps. Ainsi, on pourra choisir préférentiellement l'ordre suivant : $e_0=2^{16}+1$, $e_1=3$, $e_2=17$.

Dans une variante, on a pour tous les i, $e_i \leq 2^{16}+1$ et l'étape b) est remplacée par une autre étape de test consistant à :

si $\delta(p)=\delta(q)$, tester si la valeur e_i choisie vérifie la relation: $(1-e_i.d)$ modulo $n < 2^{(\delta(n)/2)+17}$

ou ladite relation simplifiée :

$(-e_i.d)$ modulo $n < 2^{(\delta(n)/2)+17}$

avec $\delta(p)$, $\delta(q)$, $\delta(n)$ les fonctions donnant le nombre de bits codant respectivement le nombre p, le nombre q et le nombre n;

sinon, dans le cas où p et q sont déséquilibrés, tester si la valeur e_i choisie vérifie la relation : $(1-e_i.d)$ modulo $n < 2^{g+17}$

ou ladite relation simplifiée:

$(-e_i.d)$ modulo $n < 2^{g+17}$

avec g=max(δ(p),δ(q)), si δ(p) et δ(q) sont connus ou, dans le cas contraire, avec g= δ(n)/2+t, où t désigne le facteur de déséquilibre ou une borne sur ce facteur.

Dans une autre variante, l'étape b) est remplacée par une autre étape de test consistant à :

tester si la valeur e_i choisie vérifie la relation selon laquelle:

les *premiers* bits de poids forts de $(1-e_i.d)$ modulo n sont nuls ;

ou ladite relation simplifiée selon laquelle :

les *premiers* bits de poids forts de $(-e_i.d)$ modulo n sont nuls.

De préférence, le test est effectué sur les 128 premiers bits de poids fort.

Selon un mode de réalisation préféré de l'invention, l'algorithme de cryptographie est basé sur un algorithme de type RSA en mode standard.

Selon une caractéristique, une valeur e_i ayant été attribuée à e, les calculs utilisant la valeur e consistent à :

-choisir un entier aléatoire r;

-calculer une valeur d^* telle que $d^* = d+r.(e.d-1)$;

-mettre en œuvre une opération privée de l'algorithme dans laquelle une valeur x est obtenue à partir d'une valeur y en appliquant la relation $x = y^{d^*}$ modulo n.

Selon une autre caractéristique, une valeur e_i ayant été attribuée à e, le procédé de l'invention consiste à obtenir, à l'issue d'une opération privée de l'algorithme, une valeur x à partir d'une valeur y et

les calculs utilisant la valeur e consistent à vérifier si $x^e = y$ modulo n.

De préférence, l'ensemble E comprend au moins les valeurs e_i suivantes 3, 17, $2^{16}+1$.

L'invention concerne encore un composant électronique caractérisé en ce qu'il comprend des moyens pour la mise en œuvre du procédé tel qu'il vient d'être défini.

L'invention concerne également une carte à puce comprenant un composant électronique tel que défini.

D'autres caractéristiques et avantages de la présente invention ressortiront plus clairement de la description qui est faite ci-après, à titre indicatif et nullement limitatif.

La présente invention décrit donc différentes techniques permettant de valider la valeur d'un exposant *public* e que l'on ne connaît pas a priori. Ces techniques peuvent être mises en œuvre par tout dispositif ou composant électronique doté de moyens de calculs cryptographiques adéquats, en particulier une carte à puce.

L'objet de l'invention est basé sur la constatation suivante : soit un ensemble E comprenant au moins les valeurs de e suivantes : $e_0 = 2^{16}+1$; $e_1 = 3$ et $e_2 = 17$; cet ensemble E de valeurs couvre environ 95% des valeurs des exposants *publics* couramment utilisés dans les calculs des algorithmes de cryptographie de type RSA.

La première technique proposée par la présente invention, valable pour le mode standard de l'algorithme RSA, consiste alors d'une façon générale à

choisir e_0 et à vérifier que $e=e_0$; si $e \neq e_0$ alors on essaie avec e_1; et si $e \neq e_1$, alors on essaie avec e_2.

Il se peut que pour une certaine application correspondant aux 5% d'autres cas, e ne soit pas égal ni à e_0, ni à e_1, ni à e_2. Aussi, désigne-t-on plus généralement la valeur de e par e_i. Et la méthode consiste finalement à choisir une valeur e_i parmi les e_i envisagés et à vérifier que $e = e_i$.

Plus particulièrement, la première technique pour retrouver la valeur de e, valable pour le mode standard de l'algorithme RSA, est basée sur le raisonnement suivant :

Dans le mode standard, l'algorithme privé (mettant en œuvre une opération de signature ou de déchiffrement d'un message) dispose de la valeur du module n et de l'exposant *privé* d.

Ainsi, de l'expression (1), il découle qu'il existe un entier k tel que:

$$e.d = 1 + k\,\Phi(n),$$

soit : $1-e.d = -k\,\Phi(n) = -k.(n-p-q+1)$

En réduisant les deux côtés de l'expression modulo n, on obtient :

$$1-e.d = k(p+q-1) \text{ (modulo n)}.$$

En notant que l'on a toujours $k<e$ *lorsque e est relativement petit*, l'expression précédente peut aussi s'écrire:

$$(1-e.d) \text{ modulo } n = k(p+q-1). \quad (6)$$

Le côté gauche de l'équation 6 a sensiblement la taille du module n, tandis que le côté droit a sa taille définie selon l'expression suivante quand p et q

sont équilibrés, c'est-à-dire de même taille $\delta(p) = \delta(q)$:

$$k.(p+q-1) < e.2^{(\delta(n)/2)+1}$$

avec $\delta(n)$, $\delta(p)$, $\delta(q)$ les fonctions donnant le nombre de bits codant respectivement le nombre n, le nombre p et le nombre q.

Quand p et q ne sont pas de même taille, on appelle la fonction $g=\max(\delta(p),\delta(q))$, c'est-à-dire la fonction donnant le maximum des longueurs de p et q *dans le cas où $\delta(p)$ et $\delta(q)$ sont connus; sinon, on prend $g=\delta(n)/2+t$, où t désigne le facteur de déséquilibre ou une borne sur ce facteur dans le cas contraire.* Dans ce cas où p et q sont déséquilibrés, la formule de l'expression ci-dessus devient :

$$k.(p+q-1) < e.2^{1+g}$$

En effet, comme n = p.q, si p et q sont équilibrés, alors on a l'expression $p+q<2^{(\delta(n)/2)+1}$; à l'inverse, si p et q sont déséquilibrés, alors : $p+q<2^{1+g}$

Ainsi, pour tous les e_i possibles dans l'ensemble E, si $\delta(p) = \delta(q)$, on teste si la valeur e_i choisie vérifie la relation prédéterminée suivante :

$$(1-e_i.d) \text{ modulo } n < e_i.2^{(\delta(n)/2)+1} \qquad (7)$$

sinon, on teste si la valeur e_i choisie vérifie la relation prédéterminée suivante :

$$(1-e_i.d) \text{ modulo } n < e_i.2^{g+1} \qquad (7')$$

si la relation prédéterminée de test appliquée est vérifiée, alors $e=e_i$ et on mémorise e,

sinon, on choisit une autre valeur de e_i dans l'ensemble E et on réitère les étapes précédentes.

Dans une première variante, le test pour retrouver la valeur de e:

$(1-e_i.d)$ modulo $n < e_i.2^{(\delta(n)/2)+1}$ ou

$(1-e_i.d)$ modulo $n < e_i.2^{g+1}$, suivant que p et q soient équilibrés ou non, peut être remplacé par le test suivant:

$(1-e_i.d)$ modulo $n < B$,

avec $B \geq [\max(e_i)]\ 2^{(\delta(n)/2)+1}$ dans le cas où $\delta(p) = \delta(q)$,

et $B \geq [\max(e_i)]\ 2^{g+1}$ sinon.

Dans notre exemple, on a $E=\{2^{16}+1, 3, 17\}$. Ainsi, pour tous les i, on a $e_i \leq 2^{16}+1$ et le test précédent peut donc être simplifié de la façon suivante consistant à vérifier si:

$(1-e_i.d)$ modulo $n < B$, avec $B=2^{(\delta(n)/2)+17}$ dans le cas où $\delta(p) = \delta(q)$,

et $(1-e_i.d)$ modulo $n < B$, avec $B=2^{g+17}$ sinon.

Dans une deuxième variante du test, on peut encore simplifier le test précédent en vérifiant si les bits les plus significatifs, *par exemple* les 128 bits de poids fort, de $(1-e_i.d)$ modulo n sont nuls.

Enfin, pour cette première technique, une dernière simplification consiste à déterminer la relation prédéterminée pour le test sur les e_i en démarrant avec la relation suivante:

$(-e.d)$ modulo $n = k(p+q-1)-1$

à la place de la relation (6).

Ainsi, à partir de cette simplification, on obtient pour les relations de test (7, 7'), la simplification suivante:

$(-e_i.d)$ modulo $n < e_i.2^{(\delta(n)/2)+1}$ si $\delta(p) = \delta(q)$,

et $(-e_i.d)$ modulo $n < e_i.2^{g+1}$ sinon.

Pour la première variante, on obtient le test simplifié suivant:

$(-e_i.d)$ modulo $n < B$, avec $B=2^{(\delta(n)/2)+17}$ si $\delta(p)=\delta(q)$ et $B=2^{g+17}$ sinon.

Et, pour la deuxième variante du test, on obtient le test simplifié suivant consistant à vérifier si les premiers bits de poids fort de $(-e_i.d)$ modulo n sont nuls.

Quelle que soit la variante mise en œuvre, dans sa version simplifiée ou non, si le test n'est pas vérifié pour une valeur de e_i, on choisit une autre valeur pour e_i dans l'ensemble E jusqu'à ce qu'une correspondance soit trouvée.

Si pour l'une ou l'autre des variantes qui concernent la première technique exposée ci-dessus, il n'existe pas parmi les e_i, une valeur telle que $e=e_i$, alors il reste à constater que les calculs de l'algorithme de cryptographie RSA en mode standard faisant intervenir e ne peuvent être effectués.

Par contre, lorsque la valeur de e a pu être retrouvée parmi les valeurs e_i de l'ensemble de valeurs prédéterminées E, par l'une ou l'autre des variantes, on peut alors vérifier chaque opération privée (3) de l'algorithme de cryptographie (consistant en le déchiffrement d'un message ou la génération d'une signature) en s'assurant que la valeur x obtenue à partir d'une valeur y par application de l'opération privée vérifie la relation $x^e = y$ modulo n. Si ce n'est pas le cas, *le message déchiffré ou* la signature n'est pas retourné pour éviter toute cryptanalyse.

Comme on l'a vu, une fois que l'on connaît e, le procédé selon l'invention peut également s'appliquer à une contre-mesure, notamment contre les attaques de type DPA *(et SPA)*, telle qu'elle a été décrite plus haut dans la description. Une telle méthode ainsi consiste à: choisir un entier aléatoire r; calculer une valeur d^* telle que $d^* = d+r.(e.d-1)$; mettre en œuvre une opération privée de l'algorithme dans laquelle une valeur x est obtenue à partir d'une valeur y en appliquant la relation $x = y^{d^*}$ modulo n.

Enfin, la présente invention concerne une deuxième technique pour retrouver la valeur de l'exposant e parmi un ensemble E comprenant un ensemble de valeurs e_i prédéterminées. Comme on le verra, cette technique s'applique aussi bien dans le cas du mode standard de l'algorithme RSA que dans le cas du mode CRT.

Cette technique consiste plus particulièrement à améliorer la méthode proposée dans D1. Ainsi, les étapes suivantes sont mises en œuvre :

a) définir une valeur $\epsilon = \prod_{ei \in E} ei$

telle que ϵ/e_i soit inférieur à $\Phi(n)$ pour tout e_i appartenant à E, Φ étant la fonction indicatrice d'Euler;

b) appliquer la valeur ϵ dans un calcul prédéterminé;

c) pour chaque e_i, tester si le résultat dudit calcul prédéterminé est égal à une valeur ϵ/e_i:

- si c'est le cas, alors on attribue la valeur e_i à e et on mémorise e en vue de son utilisation dans des calculs de l'algorithme de cryptographie.

- sinon, on constate que les calculs de l'algorithme de cryptographie utilisant la valeur e ne peuvent être effectués.

En mode standard, le calcul prédéterminé de l'étape b) consiste à calculer une valeur C telle que :

C = ϵ.d modulo $\Phi(n)$, d étant la clé privée correspondante de l'algorithme RSA en mode standard telle que e.d = 1 modulo $\Phi(n)$.

Par exemple, soit l'ensemble E = {e_0=3, e_1=17, e_2=2^{16}+1}, alors ϵ = $e_0.e_1.e_2$ = 3.17.(2^{16}+1).

Ainsi, avec C = ϵ.d modulo $\Phi(n)$:

Si C = 17.(2^{16}+1) = ϵ/e_0 alors e = e_0 = 3;

Si C = 3.(2^{16}+1) = ϵ/e_1 alors e = e_1 = 17;

Si C = 3.17 = ϵ/e_2 alors e = e_2 =(2^{16}+1);

Par l'intermédiaire d'un seul calcul modulaire permettant d'obtenir la valeur de C, il est donc possible de retrouver la valeur de l'exposant e parmi un ensemble E, en fonction du résultat de ce calcul.

Selon une alternative, le calcul prédéterminé de l'étape b) consiste à calculer une valeur C telle que:

C = ϵ.d modulo $\lambda(n)$, d étant la clé privée correspondante de l'algorithme RSA en mode standard mais calculée dans cette alternative modulo la fonction de Carmichael à la place de modulo la fonction indicatrice d'Euler, et donc telle que : e.d = 1 modulo $\lambda(n)$ et λ étant la fonction de Carmichael.

Dans le cas où la valeur de e a pu effectivement être retrouvée et mémorisée, les calculs de l'algorithme de cryptographie en mode standard mettant en œuvre la valeur de e consistent à parer les attaques par faute et à mettre en place une contre-mesure,

notamment contre les attaques de type DPA *(et SPA)*, et sont identiques à ceux décrits en référence à la première technique.

Dans une variante, lorsque l'algorithme RSA mis en œuvre est en mode CRT, le calcul prédéterminé de l'étape b) consiste à calculer une valeur C telle que:

$C = \epsilon.d_p$ modulo $(p-1)$, d_p étant la clé privée correspondante de l'algorithme RSA telle que $e.d_p = 1$ modulo $(p-1)$.

Ou bien encore, telle que :

$C = \epsilon.d_q$ modulo $(q-1)$, d_q étant la clé privée correspondante de l'algorithme RSA telle que $e.d_q = 1$ modulo $(q-1)$,

ou bien les deux, et à prendre le e qui nous est donné par au moins un des deux tests.

Dans le cas où la valeur de e a pu effectivement être retrouvée et mémorisée, les calculs de l'algorithme de cryptographie en mode CRT mettant en œuvre la valeur de e consistent à parer les attaques par faute.

On peut alors vérifier chaque opération privée en mode CRT de l'algorithme de cryptographie (consistant en le déchiffrement d'un message ou la génération d'une signature) en s'assurant que la valeur x obtenue à partir d'une valeur y par application de l'opération privée en mode CRT vérifie d'une part, la relation $x^e = y$ modulo p et, d'autre part, la relation $x^e = y$ modulo q.

REVENDICATIONS

1. Procédé pour la mise en œuvre sécurisée d'un algorithme de cryptographie à clé publique, ladite clé publique *étant composée d'un nombre entier n, produit de deux grands nombres premiers p et q,* et d'un exposant *public* e, ledit procédé consistant à déterminer un ensemble E comprenant un nombre prédéterminé de nombres *premiers* e_i susceptibles de correspondre à la valeur de l'exposant *public* e, caractérisé en ce qu'il comprend les étapes suivantes consistant à :

a) calculer une valeur $\epsilon = \prod_{ei \in E} ei$

telle que ϵ/e_i *soit inférieur à* $\Phi(n)$ *pour tout* e_i *appartenant à E,* Φ *étant la fonction indicatrice d'Euler;*

b) appliquer la valeur ϵ dans un calcul prédéterminé;

c) pour chaque e_i, tester si le résultat dudit calcul prédéterminé est égal à une valeur ϵ/e_i:

- si c'est le cas, alors attribuer la valeur e_i à e et mémoriser e en vue de son utilisation dans des calculs dudit algorithme de cryptographie;

- sinon, constater que les calculs dudit algorithme de cryptographie utilisant la valeur e ne peuvent être effectués.

2. Procédé selon la revendication 1, caractérisé en ce que l'algorithme de cryptographie est basé sur un algorithme de type RSA en mode standard.

3. Procédé selon la revendication 2, caractérisé en ce que le calcul prédéterminé de l'étape b) consiste à calculer une valeur C :

C = ϵ.d modulo $\Phi(n)$, d étant la clé privée correspondante de l'algorithme RSA telle que ϵ.d = 1 modulo $\Phi(n)$ et Φ étant la fonction indicatrice d'Euler.

4. Procédé selon la revendication 2, caractérisé en ce que le calcul prédéterminé de l'étape b) consiste à calculer une valeur C :

C = ϵ.d modulo $\lambda(n)$, d étant la clé privée correspondante de l'algorithme RSA telle que e.d = 1 modulo $\lambda(n)$ et λ étant la fonction de Carmichael.

5. procédé selon la revendication 1, caractérisé en ce que l'algorithme de cryptographie est basé sur un algorithme de type RSA en mode CRT.

6. Procédé selon la revendication 5, caractérisé en ce que le calcul prédéterminé de l'étape b) consiste à calculer une valeur C :

C = $\epsilon.d_p$ modulo (p-1), d_p étant la clé privée correspondante de l'algorithme RSA telle que $e.d_p$ = 1 modulo (p-1).

7. Procédé selon la revendication 5, caractérisé en ce que le calcul prédéterminé de l'étape b) consiste à calculer une valeur C :

C = $\epsilon.d_q$ modulo (q-1), d_q étant la clé privée correspondante de l'algorithme RSA telle que $e.d_q = 1$ modulo (q-1).

8. Procédé selon la revendication 5, caractérisé en ce que le calcul prédéterminé de l'étape b) consiste à calculer deux valeurs C_1 et C_2 telles que :

$C_1 = \epsilon.d_p$ modulo (p-1), d_p étant la clé privée correspondante de l'algorithme RSA telle que $e.d_p = 1$ modulo (p-1),

$C_2 = \epsilon.d_q$ modulo (q-1), d_q étant la clé privée correspondante de l'algorithme RSA telle que $e.d_q = 1$ modulo (q-1),

et en ce que l'étape de test c) consiste pour chaque e_i, à tester si C_1 *et/ou* C_2 est égal à la valeur ϵ/e_i:

- si c'est le cas, alors attribuer la valeur e_i à e et mémoriser e en vue de son utilisation dans des calculs dudit algorithme de cryptographie;
- sinon, constater que les calculs dudit algorithme de cryptographie utilisant la valeur e ne peuvent être effectués.

9. Procédé selon l'une quelconque des revendications 3 ou 4 et selon lequel une valeur e_i a été attribuée à e, caractérisé en ce que les calcul utilisant la valeur e consistent à :

-choisir un entier aléatoire r;

-calculer une valeur d^* telle que $d^* = d+r.(e.d-1)$;

-mettre en œuvre une opération privée de l'algorithme dans laquelle une valeur x est obtenue à partir d'une valeur y en appliquant la relation $x = y^{d^*}$ modulo n.

10. Procédé selon l'une quelconque des revendications 2 à 4 et selon lequel une valeur e_i a été attribuée à e, caractérisé en ce qu'il consiste à obtenir, à l'issue d'une opération privée de l'algorithme, une valeur x à partir d'une valeur y et en ce que les calculs utilisant la valeur e consistent à vérifier si $x^e = y$ modulo n.

11. Procédé selon l'une quelconque des revendications 5 à 8, et selon lequel une valeur e_i a été attribuée à e, caractérisé en ce qu'il consiste à obtenir, à l'issue d'une opération privée de l'algorithme, une valeur x à partir d'une valeur y et en ce que les calculs utilisant la valeur e consistent à vérifier d'une part, si $x^e = y$ modulo p et, d'autre part, si $x^e = y$ modulo q.

12. Procédé selon l'une quelconque des revendications précédentes, caractérisé en ce que l'ensemble E comprend au moins les valeurs e_i suivantes 3, 17, $2^{16}+1$.

13. Composant électronique caractérisé en ce qu'il comprend des moyens pour la mise en œuvre du procédé selon l'une quelconque des revendications précédentes.

14. Carte à puce comprenant un composant électronique selon la revendication 13.

15. Procédé pour la mise en œuvre sécurisée d'un algorithme de cryptographie à clé publique, ladite clé publique *étant composée d'un nombre entier n produit de deux grands nombres premiers p et q* et d'un exposant *public* e, ledit procédé consistant à déterminer un ensemble E comprenant un nombre prédéterminé de nombres *premiers* e_i susceptibles de correspondre à la valeur de l'exposant *public* e, caractérisé en ce qu'il consiste à réaliser les étapes suivantes consistant à:

a) choisir une valeur e_i parmi les valeurs de l'ensemble E;

b) *si* $\delta(p)=\delta(q)$, tester si la valeur e_i choisie vérifie la relation : $(1-e_i.d)$ modulo $n < e_i.2^{(\delta(n)/2)+1}$

ou ladite relation simplifiée :

$(-e_i.d)$ modulo $n < e_i.2^{(\delta(n)/2)+1}$

avec $\delta(p)$, $\delta(q)$ et $\delta(n)$ les fonctions donnant le nombre de bits codant respectivement le nombre p, le nombre q et le nombre n;

sinon, dans le cas où p et q sont déséquilibrés, tester si la valeur e_i choisie vérifie la relation :

$(1-e_i.d)$ modulo $n < e_i.2^{g+1}$

ou ladite relation simplifiée :

$(-e_i.d)$ modulo $n < e_i.2^{g+1}$

avec $g=max(\delta(p),\delta(q))$, *si* $\delta(p)$ *et* $\delta(q)$ *sont connus ou, dans le cas contraire, avec* $g= \delta(n)/2+t$, *où t désigne le facteur de déséquilibre ou une borne sur ce facteur;*

c) si la relation de test appliquée à l'étape précédente est vérifiée, alors e = e_i, et mémoriser e en vue de son utilisation dans des calculs dudit algorithme de cryptographie,

- si ce n'est pas le cas, réitérer les étapes précédentes en choisissant une autre valeur de e_i dans l'ensemble E jusqu'à ce qu'une valeur de e_i puisse être attribuée à e et si aucune valeur de e_i ne peut être attribuée à e alors constater que les calculs dudit algorithme de cryptographie utilisant la valeur de e ne peuvent pas être effectués.

16. Procédé selon la revendication 15, caractérisé en ce que pour tous les i, $e_i \leq 2^{16}+1$ et en ce que l'étape b) est remplacée par une autre étape de test consistant à :

si $\delta(p)=\delta(q)$, tester si la valeur e_i choisie vérifie la relation: $(1-e_i.d)$ modulo $n < 2^{(\delta(n)/2)+17}$

ou ladite relation simplifiée :

$(-e_i.d)$ modulo $n < 2^{(\delta(n)/2)+17}$

avec $\delta(p)$, $\delta(q)$ et $\delta(n)$ les fonctions donnant le nombre de bits codant respectivement le nombre p, le nombre q et le nombre n;

sinon, dans le cas où p et q sont déséquilibrés, tester si la valeur e_i choisie vérifie la relation $(1-e_i.d)$ modulo $n < 2^{g+17}$

ou ladite relation simplifiée:

$(-e_i.d)$ modulo $n < 2^{g+17}$

avec $g=max(\delta(p),\delta(q))$, si $\delta(p)$ et $\delta(q)$ sont connus ou, dans le cas contraire, avec $g= \delta(n)/2+t$, où t

désigne le facteur de déséquilibre ou une borne sur ce facteur.

17. Procédé selon la revendication 15, caractérisé en ce que l'étape b) est remplacée par une autre étape de test consistant à :

tester si la valeur e_i choisie vérifie la relation selon laquelle:

les *premiers* bits de poids forts de $(1-e_i.d)$ modulo n sont nuls ;

ou ladite relation simplifiée selon laquelle :

les *premiers* bits de poids forts de $(-e_i.d)$ modulo n sont nuls.

18. Procédé selon la revendication 17, caractérisé en ce que le test est effectué sur les 128 premiers bits de poids forts.

19. Procédé selon l'une quelconque des revendications 15 à 18, caractérisé en ce que l'algorithme de cryptographie est basé sur un algorithme de type RSA en mode standard.

20. Procédé selon l'une quelconque des revendications 15 à 19, et selon lequel une valeur e_i a été attribuée à e, caractérisé en ce que les calculs utilisant la valeur e consistent à :

-choisir un entier aléatoire r;

-calculer une valeur d^* telle que $d^* = d+r.(e.d-1)$;

-mettre en œuvre une opération privée de l'algorithme dans laquelle une valeur x est obtenue à

partir d'une valeur y en appliquant la relation $x = y^{d*}$ modulo n.

21. Procédé selon l'une quelconque des revendications 15 à 19 et selon lequel une valeur e_i a été attribuée à e, caractérisé en ce qu'il consiste à obtenir, à l'issue d'une opération privée de l'algorithme, une valeur x à partir d'une valeur y et en ce que les calculs utilisant la valeur e consistent à vérifier si $x^e = y$ modulo n.

22. Procédé selon l'une quelconque des revendications 15 à 21, caractérisé en ce que l'ensemble E comprend au moins les valeurs e_i suivantes 3, 17, $2^{16}+1$.

23. Procédé selon la revendication 22, caractérisé en ce que le choix préférentiel des valeurs e_i parmi les valeurs de l'ensemble E est effectué selon l'ordre suivant : $2^{16}+1$, 3, 17.

24. Composant électronique caractérisé en ce qu'il comprend des moyens pour la mise en œuvre du procédé selon l'une quelconque des revendications 15 à 23.

25. Carte à puce comprenant un composant électronique selon la revendication 24.

INPI
INSTITUT NATIONAL DE LA PROPRIETE INDUSTRIELLE

BREVET D'INVENTION
CERTIFICAT D'UTILITÉ
Code de la propriété intellectuelle - Livre VI

cerfa N° 11235*03

DÉPARTEMENT DES BREVETS
26 bis, rue de Saint Pétersbourg
75800 Paris Cedex 08
Téléphone : 33 (1) 53 04 53 04 Télécopie : 33 (1) 42 94 86 54

DÉSIGNATION D'INVENTEUR(S) Page N° .1./.1.

INV

(À fournir dans le cas où les demandeurs et les inventeurs ne sont pas les mêmes personnes)

Cet imprimé est à remplir lisiblement à l'encre noire

DB 113 @ W / 270601

Vos références pour ce dossier *(facultatif)*	016870 GEM1382
N° D'ENREGISTREMENT NATIONAL	0309457

TITRE DE L'INVENTION (200 caractères ou espaces maximum)

Procédé pour la mise en oeuvre sécurisée d'un algorithme de cryptographie de type RSA et composant correspondant

LE(S) DEMANDEUR(S) :

GEMPLUS

DESIGNE(NT) EN TANT QU'INVENTEUR(S) :

1 Nom		CHEVALLIER-MAMES
Prénoms		Benoît
Adresse	Rue	Résidence le Général - 14 boulevard Ganteaume
	Code postal et ville	13400 AUBAGNE
Société d'appartenance *(facultatif)*		
2 Nom		JOYE
Prénoms		Marc
Adresse	Rue	19, rue Voltaire
	Code postal et ville	83640 SAINT ZACHARIE
Société d'appartenance *(facultatif)*		
3 Nom		VILLEGAS
Prénoms		Karine
Adresse	Rue	162, Chemin de Lieutaud
	Code postal et ville	13420 GEMENOS
Société d'appartenance *(facultatif)*		

S'il y a plus de trois inventeurs, utilisez plusieurs formulaires. Indiquez en haut à droite le N° de la page suivi du nombre de pages.

DATE ET SIGNATURE(S)
DU (DES) DEMANDEUR(S)
OU DU MANDATAIRE
(Nom et qualité du signataire)

Levallois, le 31/07/2003

Jean-Paul BENTZ
99-0308